

Section: **2.2**
Adapting the Lifecycle

Description: The software engineering methodology implements well-defined processes in a lifecycle model that can be adapted to meet the specific requirements or constraints of any software project. This section provides guidelines for adapting the lifecycle processes to fit the characteristics of the project. These guidelines help ensure that there is a common basis across all software projects for planning, implementing, tracking, and assuring the quality of the work products.

The lifecycle model has built-in flexibility. All of the stages and activities can be adapted to any size and scope software engineering project. The lifecycle can be successfully applied to software development projects, software maintenance or enhancements, and customization of commercial software. The lifecycle is appropriate for all types of administrative, business, manufacturing, laboratory, scientific, and technical applications. For scientific and technical projects, adaptations to the lifecycle may be dictated by the project stakeholders or the requirements for reporting technical results in formal reports or journal articles.

Adaptations: The lifecycle can be compressed to satisfy the needs of a small project, expanded to include additional activities or work products for a large or complex project, or supplemented to accommodate security requirements. Any modifications to the lifecycle should be consistent with the established activities, documentation, and quality standards included in the methodology. Project teams are encouraged to adapt the lifecycle as long as the fundamental software engineering objectives are retained and quality is not compromised.

The following are some examples of lifecycle adaptations.

- Change the order in which lifecycle stages are performed.
- Schedule stages and activities in concurrent or sequential order.
- Repeat, merge, or eliminate stages, activities, or work products.
- Include additional activities, tasks, or work products in a stage.
- Change the sequence or implementation of lifecycle activities.
- Change the development schedule of the work products.
- Combine or expand activities and the timing of their execution.

***Adaptations,
continued:***

The lifecycle forms the foundation for project planning, scheduling, risk management, and estimation. When a lifecycle stage, activity, or work product is adapted, the change must be identified, described, and justified in the Project Plan. *Exhibit 2.2-1, Adapting the Lifecycle*, shows how stages can be combined to accommodate different size projects and software engineering techniques. *Notes* are provided throughout the lifecycle stage chapters to identify activities that have built-in project adaptation strategies. Adaptations should not introduce an unacceptable level of risk and require the approval of the system owner and other project stakeholders.

When adapting the lifecycle model, care must be taken to avoid the following pitfalls.

- Incomplete and inadequate project planning.
- Incomplete and inadequate definition of project objectives and software requirements.
- Lack of a development methodology that is supported by software engineering preferred practices and tools.
- Insufficient time allocated to complete design before coding is started.
- Not defining and meeting criteria for completing one software lifecycle stage before beginning the next.
- Compressing or eliminating testing activities to maintain an unrealistic schedule.

***Sample
Statements:***

The following are sample statements that can be used in the Project Plan to describe different types of lifecycle adaptations. The first example shows a scenario where the Feasibility Study activity will not be conducted in the Planning Stage.

A Feasibility Study will not be performed for this software project. The need for the product has been documented in several organizational reports and was included in the fiscal year long-range plans. The platform for the project is currently used for all applications owned by this organization. There are no known vendor packages that will satisfy the functional requirements described by the system owner.

**Sample
Statements,
continued:**

The following is a sample statement that shows how work products from two different stages can be combined into one deliverable.

The Functional Design and System Design documents will be combined into one design document. A Stage Exit will be conducted when the design document is completed. To reduce the risk associated with combining the two documents, the project team will develop prototype screens and reports for review and approval by the system owner/user(s) as the prototypes are developed.

The following is a sample that shows how the eight lifecycle stages can be compressed into five stages for a small project.

This project will require 4 staff months of effort to enhance an existing application. The eight stages in the lifecycle will be combined into five stages as follows: (1) Planning, (2) Requirements and Design, (3) Programming and Testing, (4) Installation and Acceptance, and (5) Maintenance.

The following deviations will occur for document deliverables:

- *A Feasibility Study and an Analysis of Benefits and Costs will not be necessary due to the restricted software and hardware platform.*
- *The Requirements Specification will be limited to the statement of enhancement requirements.*
- *The Functional Design and System Design documents will be combined into one design document.*
- *An amendment package will be developed for the existing Users Manual.*

Exhibit 2.2-1. Adapting the Lifecycle

PROJECT SIZE

LARGE

MEDIUM

SMALL

LESS

DEGREE OF PROJECT MANAGEMENT REQUIRED

MORE

Note: Iterative development and rapid prototyping are optional techniques that can be used on any size project.

▼ = Stage exit occurs at this point.

- (1) Each iteration produces working function(s) from integrated program modules.
- (2) May produce any or all of requirements, system architecture, system design.